

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Experiences and Obstacles in Industrial Applications of Intelligent Systems

Leonardo M. Reyneri¹ and Valentina Colla²

¹*Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni, Torino,*

²*Scuola Superiore Sant'Anna, TeCIP Institute, PERCRO, Pisa, Italy*

1. Introduction

Neural networks and *fuzzy systems* are well known soft computing techniques, which date back several decades since the preliminary work of McCulloch and Pitts, Grossberg, Zadeh, and dozens of other precursors. At first, the neural network was believed to be "simple and workable solution" for all the difficult problems can be dealt with, and then gave rise to a broad interest in research around the world and garnered a lot of funding. During this preliminary period, many theories have been developed, analyzed and applied.

Later, the domain of neural networks and fuzzy systems has broadened and also many other algorithms and methods have been collected under the term of *Soft Computing* and, more general, *Intelligent Systems*. These include, among others, neural networks, fuzzy logic, wavelet networks, genetic algorithms, expert systems, etc... It was then discovered that several simple problems (the so-called "toy problems") actually found very simple solutions using intelligent systems. On the other hand, difficult problems (for example, handwriting recognition and most problems of industrial relevance), still could not be completely resolved, even if intelligent systems could contribute to simplify their solution.

Today, after several decades of alternating interest of the scientific and industrial community, after the publication of tens of thousands of theoretical and practical papers, and after several attempts to apply them in a large number of application domains, intelligent systems are now reaching a rather *mature phase*. People have begun to understand the real capabilities, potentials, limitations and disadvantages, so they are on the right path towards a widespread adoption, without excessive and inappropriate enthusiasm, but also, more importantly, with a good rationale for their use.

This chapter attempts to analyze the actual level of maturity and acceptance achieved by intelligent systems and attempts to assess how, where and why they are (or can be) accepted in the industry. Note that, although the focus is on *industrial applications*, this term generally applies also to several other real-world applications such as agronomy, economics, mathematics, weather forecasting, etc.

2. Maturity level of intelligent systems

As mentioned in the introduction, all soft computing and intelligent systems techniques suffered alternating periods of *acceptance* (due to the novelty and the promising preliminary results) and *rejection* (due to the acquired awareness of limits). Hundreds of algorithms, topologies, training rules have been: i) *conceived and developed*; ii) *tested, evaluated, tuned and optimized*; iii) *temporarily or partially abandoned* ($\gg 90\%$); iv) *accepted and applied* to real problems ($\ll 10\%$).

Most of the original theories have been nearly abandoned (like, for instance, *Hopfield networks* and *Boltzmann machines*, *glass spin theories*, *stochastic networks*, etc.) either because they could not offer reasonable performance or because they were too cumbersome to use. Other theories (like, for instance, *perceptrons*, *radial basis functions* and *fuzzy systems*) eventually reached widespread acceptance, since they are more viable.

What is then at present the level of maturity of intelligent systems? This can be evaluated from a series of clues such as:

- how many theories and paradigms *have been developed altogether*. This number should be as high as possible, to ensure that no option has been forgotten;
- how many paradigms *have survived after maybe ten years*. This should be low, to minimize the knowledge one needs to learn (see section 2.2);
- the *level of acquaintance* a typical engineer has with these techniques. This should be high and it should be achieved quickly (see section 2.1);
- the *count of accepted industrial applications* should be significant (see sections 3, 4, 5).

Due to the maturity level they reached in about half a century from the preliminary works, intelligent systems now deserve to be in the *knowledge briefcase* of each engineer, economist, agronomist, scientist, etc. *together with, and at the same acceptance level of* several other basic techniques like algebra, statistics, geometry, etc.

A way to reach a widespread industrial acceptance is to avoid using statements like:

I have used/developed a neural network for...

but, instead:

I have just developed a complex system with interacting signal pre-processor, neural network, user interface, a differential equation solver, a post processor, some sensor and actuator interface, etc.

The major difference between the two approaches is which element(s) of a system receive(s) more attention by the designer. In the former statement, attention (therefore the design effort) is stressed on the presence of a neural network, which therefore improperly becomes the most relevant block. In the latter statement, the neural network takes its proper place, that is, at the same level as all the other system elements. In many cases, the blocks surrounding intelligent subsystem are the most complex to design and use.

An example for this is in the field of image processing and handwriting recognition, where a successful application relies much more on a proper image pre-processing (filtering, contrast enhancement, segmentation, labelling, skeletonization, etc.) than on the neurofuzzy processing itself.

Despite the level of maturity they reached, intelligent systems still experience a lot of difficulties to be *accepted* by the industrial community, which still sees them as *academic experiments* or *bizarre techniques* and not as a powerful tool to solve their problems. Thus, *what still misses to a complete industrial maturity?* This will be analyzed in the following.

2.1 Availability of expertise and training personnel

Knowledge and expertise on intelligent systems cannot easily be found in the industrial domain, at least in decision-making people (namely: *decision staff*, *businessmen* and *engineers*), except perhaps in the newest generation (these are still too few and not yet high enough in the decision-making stair). Decision-making people are the key actors for having intelligent systems accepted in the industry. On the other hand, the real experts in intelligent systems are those who have been trained for long time in the area of soft computing, but these often have *too little knowledge of the specific problem they are faced with*, therefore they might not tackle the problem in the most appropriate or efficient way.

Personnel training is rather time consuming, therefore costly, for industry and it can seldom be afforded unless there is a reasonable guarantee to get appropriate returns. It must be remembered that adopting *any* novel method *may* offer advantages, but it *surely* costs money. The lack of good expertise, together with people laziness often leads to using oversized networks, oversized training sets, conservative choices for paradigms and learning coefficients, etc. Altogether, more complex (therefore more costly) networks, longer design and training time, less advantages; in conclusion, *less chance of acceptance*.

2.2 The apparent diversity of neurofuzzy paradigms

At the beginning, neural networks, fuzzy systems and other soft computing techniques like wavelet networks, Bayesian classifiers, clustering methods, etc., were believed to be independent, although complementary, methods, which had to be analyzed and studied independently of each other. This caused an excessive effort to study, analyze, get familiar with a huge variety of methods and therefore to train personnel consequently. It was also believed that each paradigm had its characteristics and preferred application domains, such that a lot of experience was required to choose the best architecture for any application.

On the contrary, Reyneri (Reyneri, 1999) proved that most soft computing techniques are nothing but different *languages* for a few basic paradigms. For instance, he proved that Perceptrons, Adaline, Wavelet networks, linear transforms, and adaptive linear filters are equivalent to each other. Also Fuzzy logic, Radial Basis, Bayesian classifiers, Gaussian regressors, Kernel methods, Kohonen maps and fuzzy/hard c-means are equivalent methods, as well as Local-global networks, TKS fuzzy systems and gain scheduling controllers which are also equivalent.

With a good use of such *neurofuzzy unification*, the number of independent paradigms reduces to as few as four. All known topologies for neural, fuzzy, wavelet, Bayesian, clustering paradigms, etc. and supervised or unsupervised training algorithms are, in practice, just particular implementations and interconnections of four elementary blocks, namely: i) *computing elements*; ii) *computing layers*; iii) *normalization layers* and iv) *sensitivity layers*. All traditional neurofuzzy paradigms are then nothing else than specific *languages*, each one being more appropriate to any given application.

The efficient use of *neurofuzzy unification* would simplify personnel training, since: i) it helps the practitioner to quickly learn and get familiar with the very few basic paradigms; ii) it augments flexibility and performance of intelligent systems; iii) it therefore increases the economical return and iv) reducing the corresponding risks. Nevertheless, neurofuzzy unification is still far from being widely applied, for several historical reasons. The permanence of tens of apparently different paradigms often still creates *confusion, noise, disaffection*; it increases *personnel training costs* and reduces *advantages*; altogether it significantly reduces the *appeal of neurofuzzy systems*.

3. Relevant characteristics for industry

In this section, we try to analyze here some of the reasons why intelligent systems still experience difficulties in being accepted as an industrial standard.

3.1 Crypticity

Many intelligent systems are often felt to be rather *cryptic*, in the sense that nobody can really understand why and how a trained network solves a given problem. Apart from the many theoretical proofs that an intelligent system is capable of solving a large variety of problems, the real industrially-relevant question is that all the knowledge of a trained network is hidden within a chunk of numbers, usually arranged into *weight* or *centre matrices* or *genomes*. There is usually no clue on how to interpret such “magic numbers”, thus engineers are often sceptical in regards to their correctness, reliability or robustness.

In practice, correctness of weights is based on a successful training, although it is often difficult to either guarantee or feel that training has properly succeeded. Quality of training is measured on the amount of a residual *error measure*, but there is often no indication on which is an appropriate value for this error, especially when *sum-of-errors measures* are used, as in several commercial simulation tools. The user cannot reliably argue that a trained model is really representative of the desired system/function.

Furthermore, most training processes are often based on some amount of *randomness*, which is seldom appreciated in the industrial domain. On the other hand, traditional design methods (namely, those not using intelligent systems) are based on some predictable analytical or empirical model which is *chosen by the designer*, together with its parameters. Designer's *knowledge and experience* usually provide enough information to properly solve a problem, even though seldom in an optimal way. Nothing is apparently left to randomness.

In reality, the process of empirical adaptation of a given analytical or empirical model to a given system resembles the approach of training/adapting a soft computing system (which is nothing but a highly generic parametric model) based on a set of training data. Yet everybody considers the former as normal and straightforward, while most designers are still sceptical when facing the latter. Why is that so?

One of the reasons is that traditional (namely, non-intelligent) parametric models currently used in practice are much less generic than any soft computing models; therefore they are always under total control of the engineer, who is capable of properly interpreting parameters and values.

For instance, the model of an *electric motor* can model nothing else than an *electric motor*, and its parameters represent, for instance, winding resistance and inductance, rotor inertia, friction, etc. which are directly measurable and for which the designer can feel if they assume reasonable values or not. By appropriately varying these parameters, the model will be adapted to either *large or small motors*, either *fast or slow*, but it will never be able to model, for instance, a *chemical process*. The designer can easily become aware that, for instance, an improperly tuned model has a too large or too narrow winding resistance in comparison with the size of the motor under examination. He can therefore immediately be aware of improper tuning or of some motor fault or damage and behave accordingly.

On the other hand, intelligent systems are so generic that they can adapt to virtually any system, either electrical or chemical or economical or mechanical or agronomic, etc. The same parameters can therefore mean anything, depending on the actual use of the network (e.g. pollution of a chemical process, yield of a manufacturing process, winding resistance of a motor, rate of infection in an agricultural plant, etc.); in addition, parameters are interchangeable and there is no clue to understand what a given parameter really represents in practice. Further, nobody will ever be aware that training has not been done correctly and whether the model really represents a given system or not.

3.2 How to avoid crypticity

The use of modern unification paradigms (Reyneri, 1999) allows to easily convert neural and wavelet networks into fuzzy systems and viceversa, with several advantages, among which, for instance:

- a given neuro/wavelet network can be converted into fuzzy language, thus interpreted linguistically by experts, who are then able to “validate” and consequently “accept” an otherwise cryptic neuro/wavelet model;
- human experience, usually expressed as a set of fuzzy rules, can be converted into a neural network and then empirically tuned by means of an appropriate training set. Fuzzy (or expert systems) rules are usually understandable by an expert, such as he/she can understand the “concept” which lays behind them. An appropriate neural training of the rules therefore allows to fine tune the expert’s knowledge based on the available empirical evidence.

It is therefore mandatory to abandon all the older approaches who were more like “magic formulae” than real engineering methods and concentrate on modern approaches that consider neural, wavelet, fuzzy, Bayesian, regressor, clustering techniques, etc. as *interchangeable paradigms*. The ever lasting fight among neural- and fuzzy-people is so detrimental, as it helps to maintain the level of crypticity high, therefore preventing a widespread acceptance of intelligent systems.

The choice between, for instance, *neural networks* and *fuzzy logic* should therefore be converted into a more appropriate selection between a *neural* and a *fuzzy language*, which should be chosen depending on: i) the available knowledge from human experts; ii) the size of available training set; iii) the availability of other piece of information on the problem; iv) the level of crypticity which is accepted; v) if and how the model has to be interpreted by humans or processed by computers.

3.3 Gathering data for network training

Many intelligent systems (mostly, those based on *neural languages*) rely on the availability of *empirical data*, which is usually gathered into large *training sets*. Unfortunately, these are often too expensive to obtain, as each data point is usually an appropriate *measurement* of a mechanical or chemical or biological or economical process. Several processes are so slow that each point may require up to several days to be acquired. In some cases, if an accurate numerical model is available, computer simulations can substitute direct measurements.

Some soft computing techniques (in particular, those based on *fuzzy* and *Bayesian languages*) may require much smaller training sets, as they rely on a predefined model, described in *linguistic terms* according to *previous human experience*. This is the main reason why fuzzy logic has been accepted more quickly and extensively by industry than neural systems.

An industrial manager has to consider attentively the trade-off between the cost of gathering a large training set and the reliability of the trained neurofuzzy network. As already said, this trade-off often pushes towards the use of *fuzzy languages* whenever possible and bounds the use of *neural languages* to applications which have enough (historical) data available.

3.4 Analytical vs. empirical methods

As already said, one of the advantages of intelligent systems is that a given analytical/empirical model is by definition specific and cannot be tailored to a different problem, while neural networks are. Furthermore an analytical/empirical model usually comes after years of improvements, while neural networks are trained in a short time. Yet, a purely analytical model can be developed without any field measurement, while an empirical model requires a limited amount of field measurement. Instead most intelligent systems always require a huge amount of field measurements which, in several cases, can take years to gather.

Last but not least, the amount of field measurements which is required (that is roughly the development time) is a function of the reliability which is asked to the model. A large training set is in fact mandatory in industry to offer an adequately high reliability, while reliability of analytical models is often independent of field measurements but relies on designer's experience.

3.5 Performance is always optimistic...

Virtually any paper published in literature shows that, for a “wide range of applications, neural networks and fuzzy systems offer tremendously good performance”.

Unfortunately, more than 90% of them do not even try to afford a fair performance comparison with other *state-of-the-art* techniques and it becomes difficult to feel how good such performance really is. Just as an example, a paper (not cited) claimed that *the proposed neural model of a biochemical process is 90% accurate* and the author was *enthusiast* of that *incredible result*. Since the reviewer had little experience on modelling that specific process, he could not do anything else than blindly accept author's statement. But, when the paper was read by an experienced colleague, he pointed out that state of the art had already achieved about 95% since a few years, making those results useless for industry. It is quite

sure that the author was really convinced of the optimality of his result, due perhaps to his limited experience on the specific application domain, which was not enough to judge.

What the author surely did was to try a number of different topologies, paradigms, network sizes, training algorithms and found that his own network was offering the very best performance among all those tests. All tests were neural and no test was performed according to the state-of-the-art using traditional approaches. This method is (partially) correct to *optimize the performance of a novel intelligent system* (namely, to find which are the best choices to get the best out of it, among all possible intelligent systems), but **not** to evaluate the *appropriateness of an intelligent system for the given application*, compared against a standard one.

What was true for that specific problem, was that the proposed hybrid empirical/analytical model developed by a team of experienced engineers and biologists offered a much better performance than the best existing neural network, even for a comparable computation complexity, not considering the possible performance of the state-of-the-art. The reason for that (which happens much more frequently than one can even imagine) is that human experience, knowledge and mental capacities, which are used to develop a given “non-intelligent” model, boost so much the overall performance of a given system than even an optimal intelligent system, trained in the best way but without using the available human knowledge, cannot compensate ignoring human knowledge during its development.

3.6 How to avoid optimism...

An important step towards acceptance is to avoid unnecessary and inapplicable optimism. Any development, comparison or selection has to be *fair* and based on *real and well proven data*; never on *hypotheses*. Optimism usually tends to push the designer towards a solution which then proves less performing than originally expected, therefore convincing even more the decision-making people that intelligent systems are not yet a viable solution to their problem.

3.7 Tools and support

An important step towards industrial acceptance (as for many other industrially relevant items) is the availability of an appropriate support to the development, use, integration, conduction and maintenance of the system.

An excellent intelligent system will never be applied until its use is straightforward and user-friendly. The only chance to have an intelligent system applied is therefore the development, around the intelligent system itself together with its surrounding elements (e.g. pre-processors, postprocessors, data mining, etc.), of an appropriate user interface and development tool which supports, in the order:

- the decision-making process in helping to choose the intelligent systems instead of any other traditional system
- the preparation phase (e.g. data collection, training, tuning and testing)
- the conduction phase, namely the nominal operation of the intelligent system, when applied to the industrial process under interest
- maintenance, to overcome any problem might occur during conduction.

4. Case studies

This chapter will present a number of real industrial applications where all the aspects described and commented in the previous sections have been applied. Most of them come from our personal experience, as acquiring enough, reliable and trustable details from other people is usually difficult.

4.1 Prediction of Jominy profiles of steels

The *Jominy profile* of a steel is a curve obtained in a test, where a small cylindrical specimen of steel is kept at a very high temperature (usually more than 1500 °C) and one end of the specimen is cooled by quenching it for at least 10 min. in a water stream, while the other specimen end is cooled in air. This treatment causes a cooling rate gradient to develop over the length of the specimen, with the highest cooling rate corresponding to the quenched end. This procedure affects the steel micro-structure along the length of the specimen and, as a consequence, the steel hardness in the diverse portions of the test bar. The Jominy profile is built by measuring the specimen hardness values h_i on the Rockwell C scale at increasing distances d_i from the quenched end. Several studies investigated the correlation between the shape of such curve and the steel chemistry (Doane & Kirkaldy, 1978) and some of them applied neural networks to this purpose, such as (Vermeulen et al., 1996).

In particular Colla et al. (Colla et al., 2000) propose a parametric characterization of the profile, namely the approximation of the generic profile with a parametric curve, and then predict the shape of each profile through a neural network which links the steel chemistry to the curve parameters.

This approach proved to be successful when the “shape” of the profile is constant (which happens, for instance, when dealing with the same steel grades produced by one single manufacturer). On the other hand, when facing the prediction of the Jominy curve of many different steel grades manufactured by different steel producers, the actual shape of the curve might considerably vary and the parametric approach is no more successful.

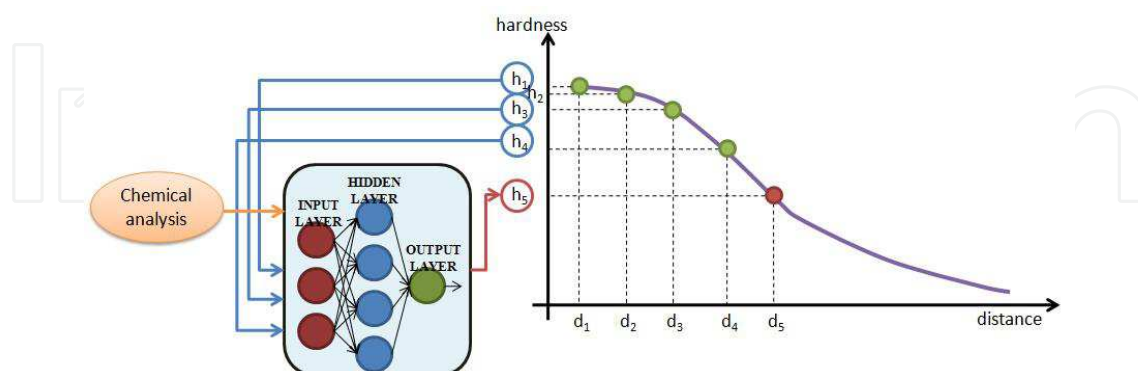


Fig. 1. Conceptual scheme of the sequential predictor of Jominy curves.

In fact, a different approach to the same problem has been proposed by Marin et al. (Marin et al. 2007), where a neural sequential predictor has been proposed: here, apart from the first two points of the curve (i.e. the ones corresponding to the lowest distance values from the quenched end), each single point of the Jominy profile is singularly predicted by a neural

network having as inputs the contents of some chemical elements and some of the previously predicted hardenability values, according to the schematic description provided in Fig. 1.

This application proved to be successful because: i) results were accurate and accuracy could easily be measured; ii) the intelligent systems (a neural network in both cases) was very simple, with few weights, and these could easily be interpreted by the technician; iii) the neural predictor has been coupled with a user-friendly software interface allowing not only to run the model, but also to collect data and re-train all the neural networks with new data provided by the user, so that each steel company can progressively “specialise” the predictor on its own steel grades; iv) training time for using the software tool which was developed was very short. It is worth noting that, as pointed out at the beginning, the neural network itself is just a small element of the whole system (software tool, pre-processing, data collection, result presentation, etc.

4.2 Prediction of malfunctioning during steel casting

In the standard steelmaking practice, during continuous casting, the liquid material produced in the blast furnace is cast, after some manufacture, into the ladle and, subsequently, into the tundish (see Fig. 2). On the bottom of the tundish, some nozzles are located, through which the liquid steel passes into the mould or strip casters. The section of such nozzles is far smaller with respect to the tundish dimensions. When particular steel grades are produced, some alumina precipitation on the entry and on the lateral surface of these nozzles can partially or even totally block the flow of the liquid steel. This phenomenon is commonly known as *clogging* and is highly detrimental to casting reliability and quality of the cast products.

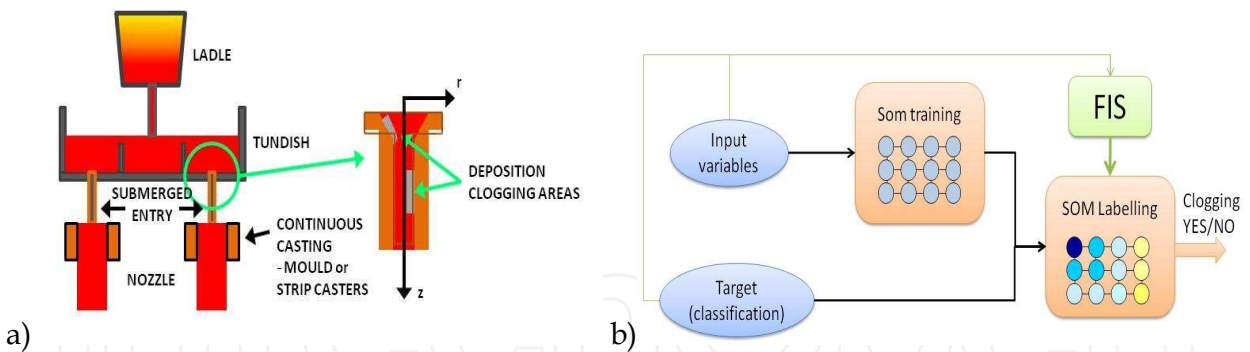


Fig. 2. a) Location of the nozzles that can be occluded; b) Schematic description of the labelled SOM-based classifier.

The clogging phenomenon is still not deeply understood (Heesom 1988), due to the very high number of chemical and process factors affecting the occurrence of the precipitation of the materials on the nozzle internal surface as well as to the impossibility of installing complex systems of probes and sensors in order to closely observe the phenomenon itself.

For this reason, some attempts have been performed to apply intelligent systems for the prediction of clogging occurrence on the basis of the steel chemical composition and on the process parameters. In particular, such as it can be frequently found in fault diagnosis applications, the prediction of clogging has been faced as a binary classification problem

where one of the two classes to be distinguished (i.e. the one corresponding to malfunctioning) is far less frequent than the other one. Firstly Self Organising Maps (SOM) have been applied (Colla et al., 2006) in order to predict the clogging occurrence, in parallel with a physical model that takes into account some basic mechanism of the alumina precipitation and the geometry of the nozzles, but is not capable to explain all the complex relationships between process and chemical variables.

The performance of the overall system are acceptable (this is also a key element in industry: the aim is seldom to optimize but often to achieve any performance better than a given threshold in a limited time) and the system has been successfully applied in the industrial context, mainly because: i) after a short testing, the prediction accuracy was proven to be higher; ii) risk was little as the traditional approach could be used to crosscheck the predictions of the intelligent system; iii) the availability of a simplified end-user interface reduced personnel training to the minimum, allowing the operator to input the relevant process parameters and obtain immediate indication of the actual danger of clogging occurrence and the potential countermeasure to adopt (i.e. Calcium Oxide addition to the liquid steel) (Fera et al. 2005).

Improvements are also possible, by taking into account the different importance of misclassification errors. In fact the erroneous classification of a faulty situation as a normal one (sometimes called *missed detection*) prevents the operators to develop suitable countermeasures to avoid the clogging, with potential heavier consequences with respect to the opposite case, when a potential unnecessary warning message is raised in a standard condition (the so-called *false alarm*). Actually standard classifiers are not always capable of providing excellent results when dealing with imbalanced datasets: therefore in (Vannucci & Colla 2011) a classifier has been applied, which is explicitly designed to cope with imbalanced datasets and exploits the labelled SOMs, according to the scheme depicted in Fig. 2.b. Once trained, each neuron of the SOM is labelled as corresponding to the frequent or infrequent class through a procedure that exploits a Fuzzy Inference System in order to find a suitable compromise between missed detection and false alarms rates.

4.3 Prediction of the end-point in the converter

In the integrated steelmaking cycle, where steel is produced from primary raw materials, the Blast Oxygen Furnace (BOF) is the plant where steel is produced from pig iron, scrap and lime, by blowing oxygen to burn off the carbon. In the BOF, the sub lance device that measures carbon content and temperature rapidly before the late period of blowing is the most important tool for BOF process control. The use of such sub lance has been an important step for controlling the BOF steel making processes. Sub lance is basically used to take sample at the end of blow usually 3-4 minutes before end of blow for analysis of sample and also measures temperature of the bath. Since the introduction of sub lance, the accuracy of the end point prediction (hit rate) at most of steel plants has gradually increased from approximately 60% to 90%. In (Valentini et al. 2004) a neural network has been applied to predict the final Carbon content $[C]$ for an OBM (from the German *Oxygen Bodenblasen Maxhuetten*) converter in steel making industry by exploiting the estimates of the Oxygen content $[O]$ and of the temperature T . These three variables are usually linked by the following approximate mathematical equation:

$$\text{Log} \frac{P_c}{[C]^n [O]} = \frac{A}{T} + B \quad (1)$$

where P_c is a constant pressure value in the range [1, 1.5] atm, A and B are constants whose nominal values are respectively, $A=1895$ °K and $B=1.6$ and n is commonly assumed to be unitary, but some literature results provide $n \approx 0.5$ for $[C] < 0.08\%$ and $n \approx 1$ otherwise.

Equation (1) can easily be inverted in order to predict $[C]$ from $[O]$ and T , but the prediction obtained using the nominal values of the constant parameters is not very reliable compared to the $[C]$ measurements contained in a dataset provided by the steel manufacturer for the steel grades of interest. A reliable prediction of the final Carbon content at the end of the refining process is very important, as it allows to evaluate the process parameters (such as the amount of inflated Oxygen and the duration of the refining process) required to achieve the desired results optimizing the time and cost of the production process. By adopting a simple two-layer MLP with 3 neurons in the hidden layer, the prediction error has been reduced of 64% with respect to the prediction obtained through eq.(1).

This system presents the following advantages: i) the performance is acceptable, ii) the neural network is very simple; iii) the training time is negligible. However the neural model has been not very well perceived by the end-users mainly because it is difficult to attribute a precise physical meaning and interpretation to the network parameters, such as it happens for the parameters of the formula (1).

Therefore the alternative solution of a fine tuning of the physical parameters around the nominal values has finally been preferred.

4.4 Prediction of the time required by each stage of hot rolling mills

The efficiency and productivity of steel hot rolling mills is heavily affected by the possibility of precisely estimating when the different manufacturing stages are completed, as this avoids bottlenecks and provides important time and energy savings. For this reason, several Mill Pacing Control (MPC) systems have been realised and implemented, which allow optimising the production flow starting from the reheating furnaces, where slabs are heated at a temperature between 1100°C and 1300°C for optimal workability before being rolled. Hot steel rolling mills are usually composed of two main stages, namely the *roughing mill*, where the slab is firstly compressed, and the *finishing mill*, where the aimed thickness of the hot rolled coil is reached. A further rolling stage can be afterwards required, named *cold rolling*, which is pursued at far lower temperatures in order to produce flat products, such as plate, sheets or coils of various thicknesses.

MPC systems allow shorten the discharging interval between two subsequent slabs avoiding collisions. To this aim, schedule systems are developed and simulations are performed in order to test new strategies without affecting the production cycle.

Colla et al. (Colla et al., 2010) applied neural networks to solve a particular mill pacing problem, different from the usual one, namely the prediction of the total roughing time and of the time required for passing the first gauge of the finishing mill. This investigation has been pursued in order to increase the rolling efficiency and decrease the total rolling time. The slabs that are subsequently rolled can differ in steel grade and other features, thus the related rolling processes can require different times and energy amounts. The time required

for the roughing process is in average (but not always) smaller than the finishing time. Thus a slab can be output by the roughing mill while the rolling of the previous coil is being completed: this fact may cause a collision or may force the second slab to remain stuck while its temperature decreases, which makes its successive rolling more difficult. On the other hand, the time between the input of two successive slabs to the roughing time cannot be excessive, in order to keep productivity and avoid energy losses. Ideally, a slab should be input to roughing mill exactly at a time instant that will allow it to arrive at the entrance of the finishing mill when the rolling of the previous coil is just terminated. (Colla et al. 2010) applied various neural networks-based approaches to predict the time τ_i ($1 \leq i \leq 6$) required by the slab to pass each one of the 6 stages that form the roughing mill (see Fig. 3).

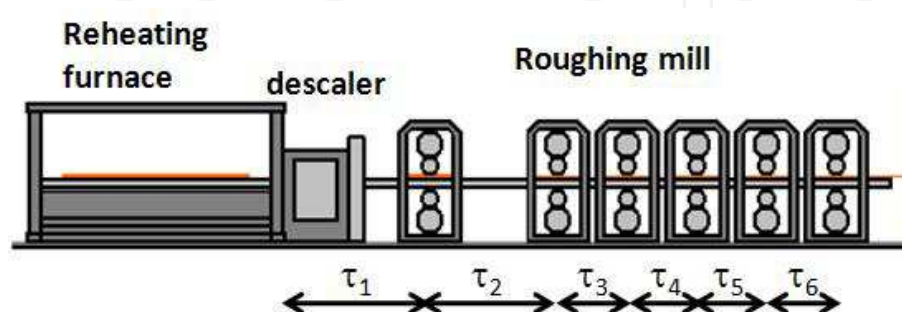


Fig. 3. Scheme of the first stages of a steel hot rolling mill.

In particular, the most successful solution performs a sequential prediction, namely bases the prediction of τ_i (for $i > 1$) not only on product and process parameters, but also on the prediction of the times required to pass the previous stages, i.e. τ_k with $1 \leq k \leq i$. Moreover, neural networks have been applied also to predict the time required for passing the finishing mill.

In this case, the application of neural networks were actually advantageous for the following main reasons: i) neural networks proved to outperform more traditional approaches; ii) the neural system is naturally adaptable to the changing operating conditions thanks to its capability of self-learning from data. However the on-site real-time implementation has not been easy and required considerable efforts because it has been difficult to interface the system with the control system of the mill.

4.5 Estimate of train position and speed from wheels velocity measurements

Within an Automatic Train Protection (ATP) system, two subsystems are usually included: a ground subsystem, which provides updated information on the train position and the line gradient by exploiting fixed balises or another source of absolute information (e.g. GPS), and an on-board subsystem, which estimates the actual train position and speed, according to the scheme depicted in Fig.4.

The ground subsystem communicates to the on-board one the distance from the next reference point on the line, the gradient of the line and the allowed speed at the next reference point. The on-board subsystem then evaluates the distances from the next information point and the minimum distance that allow compliance with the speed limit at the next reference point. If it turns out that the train cannot meet the target speed at the next reference point (as the residual braking resources of the train are not sufficient), the on-

board subsystem actuates suitable countermeasures, such as emergency breaking. The evaluation of the above-mentioned distance values requires the knowledge of the breaking parameters and of the actual train speed: a correct estimate of this last variable even in poor adhesion conditions (i.e. when one or more train wheels are sliding on the rails and, thus, the axle angular velocity is not proportional to the train speed) is crucial.

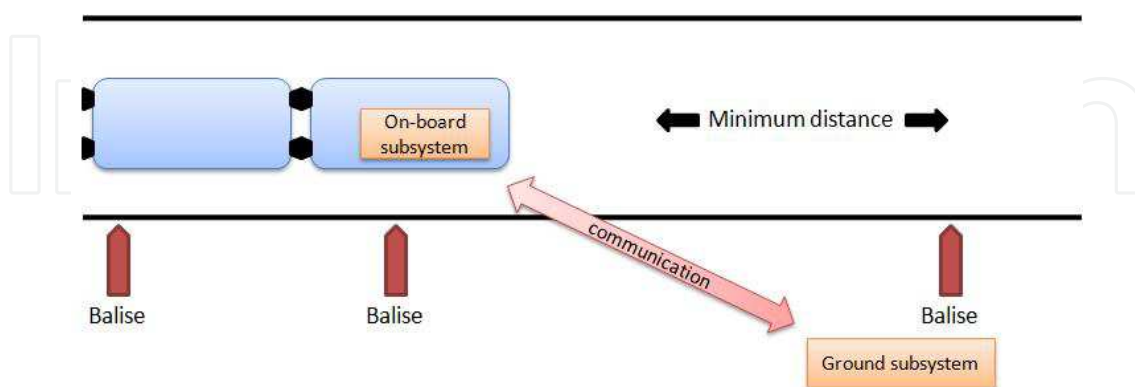


Fig. 4. Working principle of an ATP

Allotta et al. (Allotta et al 2001, Allotta et al 2002) developed a series of algorithms for estimating the actual train speed on the basis of the information collected concerning two axles of the locomotor. A first set of such algorithms have been developed according to expert personnel specifications and following the traditional “crisp” reasoning, which exploits different simple deterministic formulas for calculating the train speed depending on the condition of adhesion of the wheels to the rails. In fact, among a huge number of state variables that are considered in the procedure, there are two binary variables indicating the adhesion condition of each axle. The technical personnel of the train society formalised the reasoning that leads the human operators to a correct determination of the adhesion conditions. Then two identical fuzzy systems have been developed, which take two inputs each, namely the difference between the velocities of the two controlled axles and the acceleration of the axle whose adhesion condition is estimated, and return the degree of adhesion of one axle. The design of two fuzzy systems have been refined by means of a training procedure exploiting a great quantity of the available data and, finally, they have been merely substituted to the old crisp algorithm for adhesion condition estimation, by leaving the rest of the speed estimation procedure unchanged. As an alternative, the standard rule-based system merely implementing the human operators’ reasoning has been implemented and its own parameters (such as thresholds) have been tuned by means of Genetic Algorithms (GA) exploiting the available experimental data and adopting as fitness function to minimise the error between the actual and estimated train speed. A second set of algorithms that have been tested for this application perform a direct estimate of the train velocity taking as inputs some of the available state variables (in particular axles velocities, accelerations and acceleration variations). Both neural networks and fuzzy inference systems have been tested to this purpose.

From a comparison among all the tested approaches it turned out that the algorithms purely based on AI techniques (and, in particular, the neural network (Colla et al. 2003)) outperform the rule-based ones and have also a simpler structure. However, these systems also present the following non negligible disadvantages: i) a difficult physical interpretation of fuzzy rules or of the neural network;

ii) a difficult implementation of the specification requirements; iii) the FIS-based methods are also computationally less efficient; iv) when testing the more frequent fault conditions, all the developed algorithms present an acceptable degree of reliability and robustness, but the crisp algorithm, which is actually adherent to the specifications provided by the expert personnel of the train society, provides the best guarantee of estimated speed values lying within acceptable limits; v) with respect the merely crisp algorithm and to its improved version, where some parameters have been optimized through GA, the soft computing-based algorithms provide less control over the internal parameters of the estimator, which increase in number but loose in physical meaning (this is especially true for the neural predictor, which has been applied as a black-box parametric estimator); vi) soft computing-based procedures, and especially the ones which exploit neural networks, do not guarantee that a particular input pattern (or series of input patterns) will lead to unacceptable velocity estimates.

Thus finally the rule-based algorithm tuned through GA has been preferred to all the other approaches for the final implementation.

4.6 Optimisation of the logistic in an automatic warehouse of steel tubes

Optimisation of logistic is one of the fields where intelligent systems have most successfully been applied. Colla et al. (Colla et al., 2010) tested several AI-based techniques for the optimisation of products allocation in an automatic warehouse of steel tubes. The warehouse has been designed to stock a large variety of typologies of steel tubes, differing in quality of the steel, in the length as well as in the shape and dimensions of the section. As soon as the tubes are produced, they are grouped in packs and automatically transferred to a stocking area, where they are located in piles that must be composed by the same typology of tubes. A non optimal allocation strategy can cause the available space in the warehouse not to be fully exploited, such as, for instance, the case in which many short (i.e. composed by a few packs) piles are present in place of a few higher ones. To this aim, firstly some Key Performance Indicators (KPIs) have been defined in order to derive objective functions to be optimised by the different allocation strategies. Afterwards, an optimisation problem has been formulated, for which an analytical model of the problem was really too complex to define and implement, due to the variability of the workload and to the interaction between the automatic tube conveyors, as traffic control is only partially centralised (for instance collision avoidance is managed at local level through suitable sensing and communicating devices mounted on each conveyor). Traditional derivative-based optimisation models cannot be applied, while GAs are a very suitable solution for the optimisation problem, as they allow a decoupling between the problem formulation and the search procedure. The destination of each tubes pack has been suitably codified in a chromosome and GAs have been applied in order to minimise a fitness function obtained from a composition of the above-defined KPIs. Different ways to aggregate the selected KPIs have been tested, from a simple weighted sum up to a Fuzzy Inference System implementing a complex combination according to rules derived from the knowledge of the technical personnel working on the plant. However, this application is intrinsically a Multi-Objective Optimization (MOO) problem, as the KPIs represent requirements that are often in contrast to each other. Any kind of aggregation of the KPIs simplifies it to a Single Objective Optimization problem, but surely the most suitable way to cope with this problem is by exploiting GA-based MOO algorithms. The Strength Pareto Evolutionary Algorithm (Zitzler & Thiele 1999) has been successfully applied to this problem and outperforms all the other approaches.

The success of this applications of intelligent systems with respect to the previous system which was based on heuristics depends on the following reasons: i) a the correct formalisation of the MOO problem; ii) a suitable simulation system of the automatic warehouse (Colla & Nastasi 2010) that has been realised in order to reproduce the monitoring system of the warehouse itself and can be fed with the same data files that are used by the real system; iii) the possibility (as a consequence of point ii) to easily test the different strategies in a realistic way without affecting the normal operations of the warehouse; iv) the possibility (as a consequence of point ii) of performing an easy and user-friendly comparison of the standard and simulated situation of the warehouse obtained through the previous and improved strategies is possible, which can be of help for the technical personnel in order to evaluate the advantages of a new strategy; v) the easiness of collecting training data, which are no more than standard system data; vi) the modularity of the software for simulation and for the implementation of the allocation strategy, which makes the substitution of the new code within the control system of the warehouse straightforward.

5. Conclusion

According to authors' personal experience, it *cannot* be stated that intelligent systems are so advantageous with respect to traditional techniques to be *universally* accepted for industrial applications. Or better, advantages exist but they are often too limited when compared with the additional risks, training costs, design time, and documentation/maintenance effort. There are surely applications where they provide advantages, especially in tough problems, but these are rather limited, therefore they do not justify a universal acceptance.

Unfortunately, in most industrial applications that we have encountered so far, very few intelligent system offered such better performance with respect to other techniques to really convince the sceptical user. Of course the comparison is made between the *best* intelligent system and the correspondingly *best* non-intelligent technique.

5.1 A global advantage of intelligent system

There is perhaps a major advantage which makes intelligent systems attractive in a wider range of applications. In practice, intelligent systems are:

generic approximation and modelling techniques which allow accurate system modelling/forecasting/approximation/classification/etc. without any specific experience of the designer.

In practice anybody without any experience in a specific subject can afford solving a problem which could otherwise (namely, with traditional techniques) be afforded only by an expert (or a team of experts) in that field. It is likely that an expert, with appropriate knowledge of the problem and of a bunch of more specific methods would achieve a better result, but this would be far more expensive for an industry, both because of the higher cost of the expert and for the longer development time. This is a rather interesting advantage, even when intelligent systems are suboptimal, as it significantly reduces training costs of inexperienced personnel.

5.2 How to help industry accepting intelligent systems

The authors personally believe that industry strongly needs to be helped to accept intelligent systems and this should be a major role for universities and research institutions.

Yet this has to be done in the most appropriate way, that is, by showing industry *unambiguously if, where and when* intelligent systems offer significant advantages or, more realistically, more advantages than drawbacks and associating the intelligent system with an appropriate *development environment* and enough *supporting tools*, which is often the most time consuming element to be developed.

This is one of the major reasons for the several Special Sessions on Industrial Applications of Intelligent Systems which have been held in the last decade. Authors are usually requested to present their ideas on intelligent systems but, more important, to prove that they are either comparable or significantly better than other standard techniques. *Such a comparison has to be as fair as possible, as it is not normally the case.* In practice, in most papers, intelligent systems are usually compared among themselves. The expert reader is left with the question:

Are you sure that other techniques would not be even better or simpler?

Or, when a comparison is attempted with standard techniques, these are usually much older, that is, the paper demonstrates, for instance, that an *up-to-date neurofuzzy network* is much better than an *older-than-my-father standard technique*, which is rather obvious, as technology keeps improving, independent of intelligent system.

One of the major reasons for this lack of fair comparisons is that comparing an intelligent system against an *up-to-date standard method* requires developing by scratch an appropriate demonstrator, which often requires either a lot of specific experience or a lot of time, and usually nobody wants to afford it.

Only those research groups who tightly cooperate with an industrial group can merge industrial and academic experiences, to develop both techniques appropriately, although these are seldom done together, due to unaffordable additional costs.

5.3 A critical question

So far very few applications of intelligent systems provided such better performance with respect to other techniques to really convince even the most sceptical user. In most cases, they can either offer a slightly better performance (when compared with an alternative well-designed method) with a shorter design time but, on the other hand, design risks are often so critical that they definitely impair the advantages. It is therefore time for a critical question:

In which applications are neural networks have fuzzy logic a higher chance of being accepted?

We think that, at present, the most promising areas are, for instance:

- *data mining, knowledge based systems*, where information, data, knowledge and models are *valuable items*, but they are often *hidden* in a huge amount of noise, ambiguous, contradicting data. Data is so wide, contradicting, ambiguous, that no method can be accurate and predictable, therefore neural networks may provide advantages, without the need to be 100% correct;
- *prediction/classification of partially random processes*, like *time-series prediction, forecasting, complex pattern classification, semantic Web*, etc., where the randomness of the

process/patterns prevents a 100% prediction accuracy, therefore the errors of the intelligent systems can be accepted at no cost;

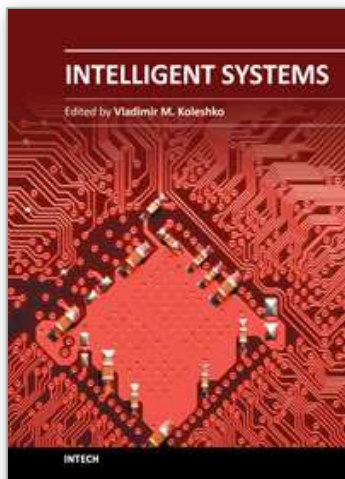
- *modelling of complex systems*, where any other modelling technique would be as incomprehensible as a neurofuzzy model;
- *consumer applications* where the appeal of the “fuzzy label” increases the market of an appliance.

6. References

- Allotta, B; Malvezzi, M.; Toni, P.; Colla, V. (2001). Train speed and position evaluation using wheel velocity measurements, *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics AIM'01*, Como, Italy.
- Allotta, B; Colla, V.; Malvezzi, M. (2002). Train Position And Speed Estimation Using Wheel Velocity Measurements *Journal of Rail and Rapid Transit Proceedings of the Institution of Mechanical Engineers Part F*, Vol. 216, No. 3, pp. 207-225.
- Colla, V.; Reyneri, L.M.; Sgarbi, M. (2000). Parametric Characterization of Jominy Profiles in Steel Industry, *Integrated Computer-Aided Engineering*, Vol. 7, pp. 217-228.
- Colla, V.; Vannucci, M.; Allotta, B.; Malvezzi, M. (2003). Comparison of traditional and neural system for train speed estimation, *Proceedings of the 11th European Symposium on Artificial Neural Networks ESANN 2003*, Brugges, Belgium, 23-25 April.
- Colla, V.; Vannucci, M.; Fera, S.; Valentini, R. (2006). Ca-treatment of Al-Killed steels: inclusion modification and application of Artificial Neural Networks for the prediction of clogging, *Proceedings of the 5th European Oxygen Steelmaking Conference EOSC'06*, 26-28 June 2006, Aachen, Germany, pp. 387-394.
- Colla, V.; Vannucci, M.; Valentini, R. (2010). Neural network based prediction of roughing and finishing times in a hot strip mill, *Revista de Metalurgia*, Vol. 46, No 1, pp. 15-21.
- Colla, V.; Nastasi, G. (2010). Modelling and Simulation of an Automated Warehouse for the Comparison of Storage Strategies, Chap.21 in *Modelling, Simulation and Optimization*, INTECH pp. 471-486 (ISBN 978-953-7619-36-7).
- Colla, V.; Nastasi, G.; Matarese, N.; Reyneri L.M. (2010). GA-Based Solutions Comparison for Storage Strategies Optimization for an Automated Warehouse” *Journal of Hybrid Intelligent Systems*, Vol. 7 pp. 283-297.
- Doane, D.V.; Kirkaldy, J.S. (1978). Hardenability Concepts with Applications to Steel, TMS-AIME, Warrendale.
- Fera, S.; Harloff, A.; Roedl, S.; Mavrommatis, K.; Colla, V.; Santisteban, V.; Roessler S. (2005). Development of a model predicting inclusions precipitation in nozzles based on chemical composition and process parameters such as casting rate, liquid temperature, nozzle design and slag composition, *European Commission Ed. Technical Report EUR 21442*.
- Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, Mc Millan College Publishing Company, New York, 1994.
- Heesom, M.J. (1988). 'Physical and chemical aspects of nozzle blockage during continuous casting, *Proceedings of the 1st Int. Calcium Treatment Symposium*, London (UK).
- Marin, B.; Bell, A.; Idoyaga, Z.; Colla, V.; Fernández, L.M. (2007). Optimization of the influence of Boron on the properties of steels, *European Commission Ed. Technical Report EUR 22446*.

- Reyneri, L.M. (1999) Unification of Neural and Wavelet Networks and Fuzzy Systems, *IEEE Trans. on Neural Networks*, Vol. 10(4), pp. 801-814.
- Valentini, R.; Colla, V.; Vannucci, M. (2004). Neural predictor of the end point in a converter", *Revista de Metalurgia*, Vol 40, No. 6, pp. 416-419.
- Vannucci, M.; Colla, V. (2011). Novel classification methods for sensitive problems and uneven datasets based on neural networks and fuzzy logic, *Applied Soft Computing*, Vol. 11, pp. 2383-2390.
- Vermeulen, W.G.; Van Der Wolk, P.J.; De Weijer, A.P.; Van Der Zwaag, S. (1996). Prediction of Jominy Hardness Profiles of Steels Using Artificial Neural Networks," *Journal of Materials Engineering and Performances*, Vol. 5, No. 1, pp. 57-63.
- Zhang, Q. (1997) Using Wavelet Network in Non-parametric Estimation, *IEEE Transactions on Neural Networks*, Vol. 8(2), pp. 227-236.
- Zitzler, E.; Thiele, L. (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Evolutionary Algorithm, *IEEE Transactions on Evolutionary Computation*, Vol. 3 (4), pp. 257-271.

IntechOpen



Intelligent Systems

Edited by Prof. Vladimir M. Koleshko

ISBN 978-953-51-0054-6

Hard cover, 366 pages

Publisher InTech

Published online 02, March, 2012

Published in print edition March, 2012

This book is dedicated to intelligent systems of broad-spectrum application, such as personal and social biosafety or use of intelligent sensory micro-nanosystems such as "e-nose", "e-tongue" and "e-eye". In addition to that, effective acquiring information, knowledge management and improved knowledge transfer in any media, as well as modeling its information content using meta-and hyper heuristics and semantic reasoning all benefit from the systems covered in this book. Intelligent systems can also be applied in education and generating the intelligent distributed eLearning architecture, as well as in a large number of technical fields, such as industrial design, manufacturing and utilization, e.g., in precision agriculture, cartography, electric power distribution systems, intelligent building management systems, drilling operations etc. Furthermore, decision making using fuzzy logic models, computational recognition of comprehension uncertainty and the joint synthesis of goals and means of intelligent behavior biosystems, as well as diagnostic and human support in the healthcare environment have also been made easier.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Leonardo M. Reyneri and Valentina Colla (2012). Experiences and Obstacles in Industrial Applications of Intelligent Systems, Intelligent Systems, Prof. Vladimir M. Koleshko (Ed.), ISBN: 978-953-51-0054-6, InTech, Available from: <http://www.intechopen.com/books/intelligent-systems/experiences-and-obstacles-in-industrial-applications-of-intelligent-systems>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen